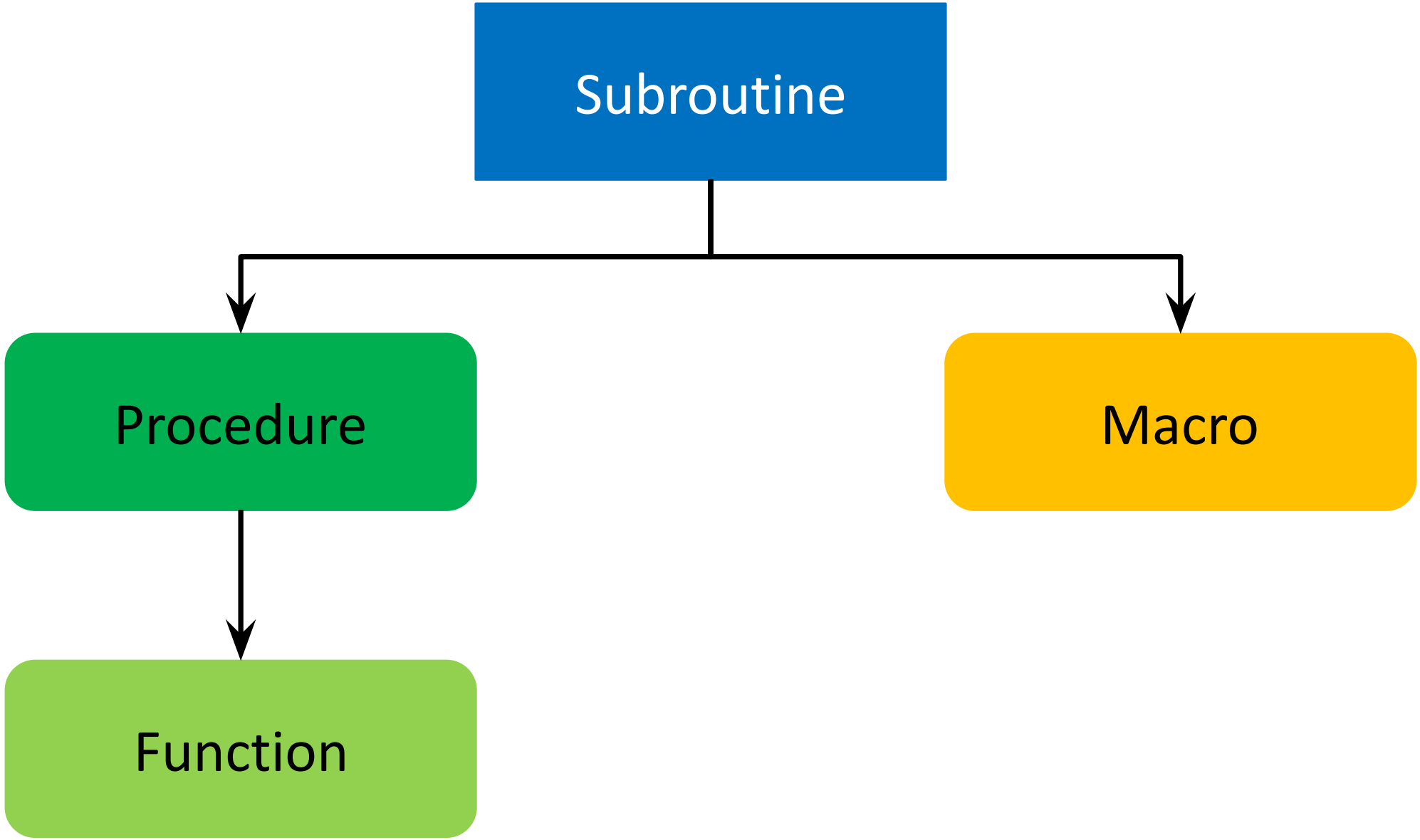
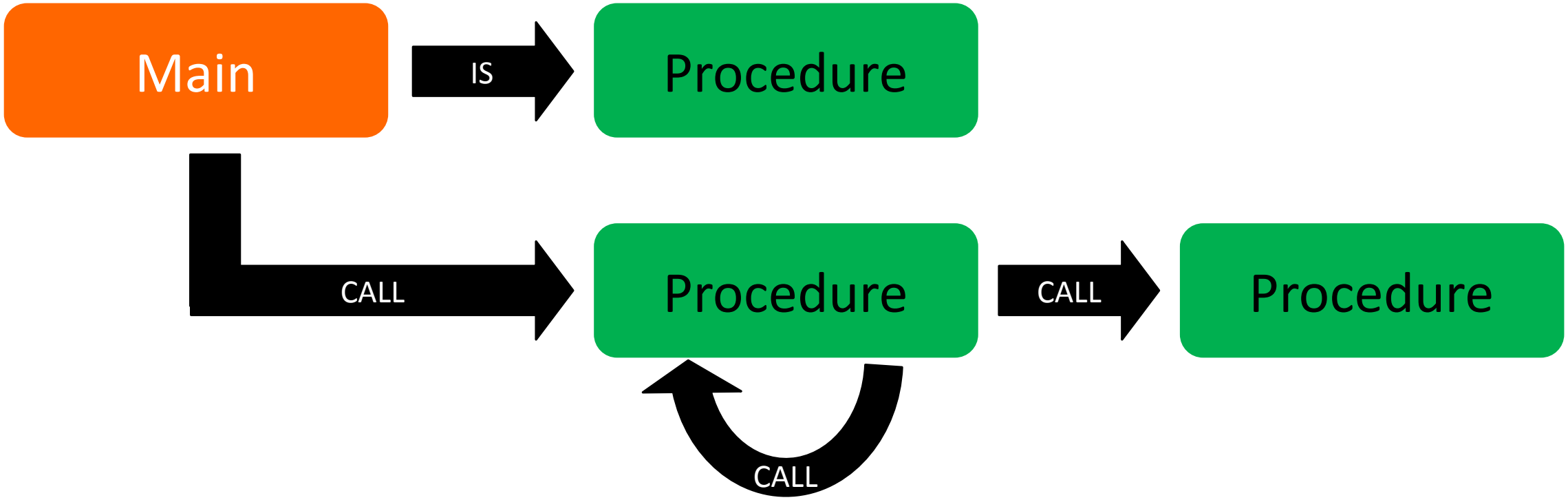
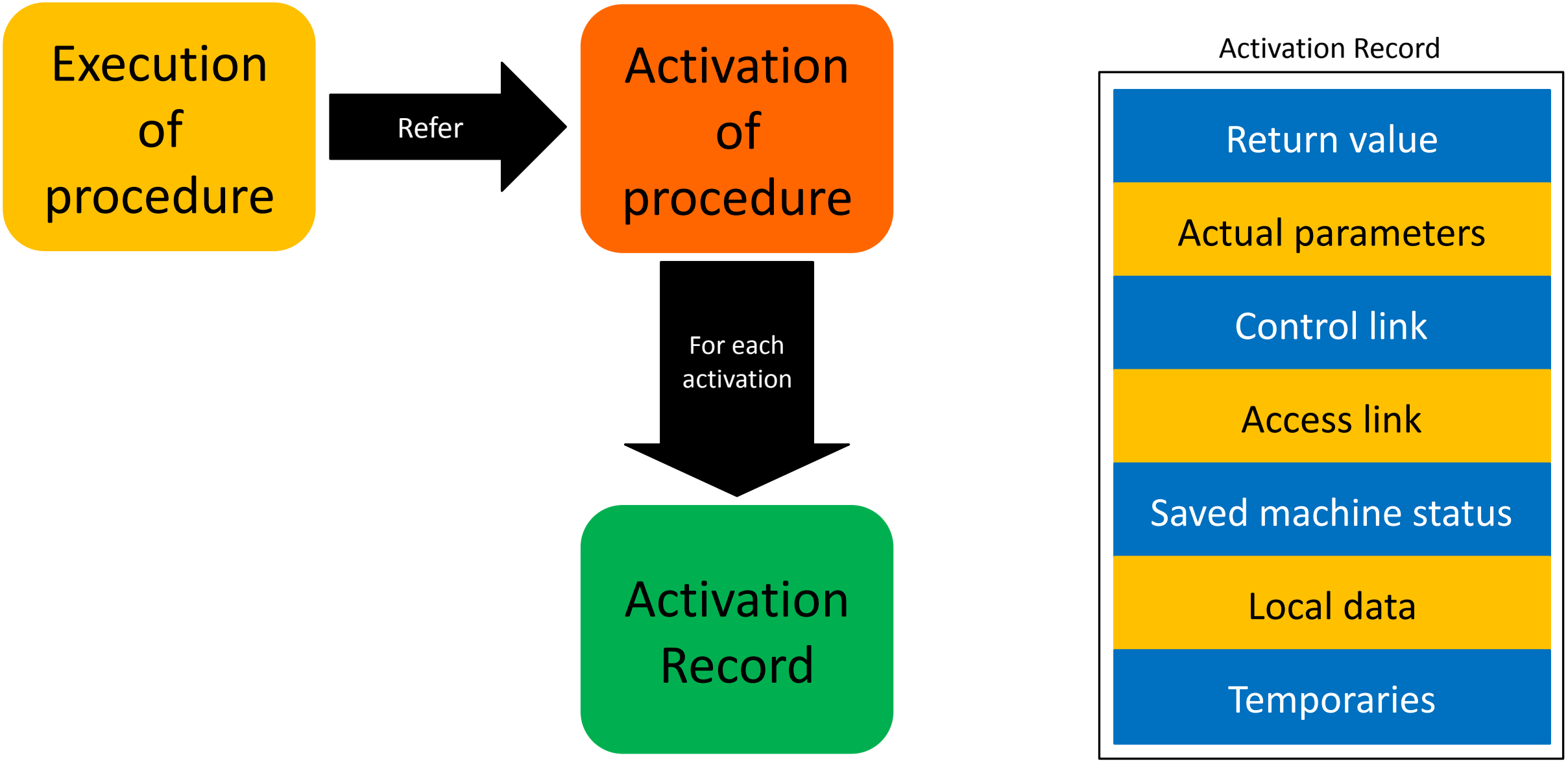
Memory Allocation & Memory Management

Ms. Trusha Patel

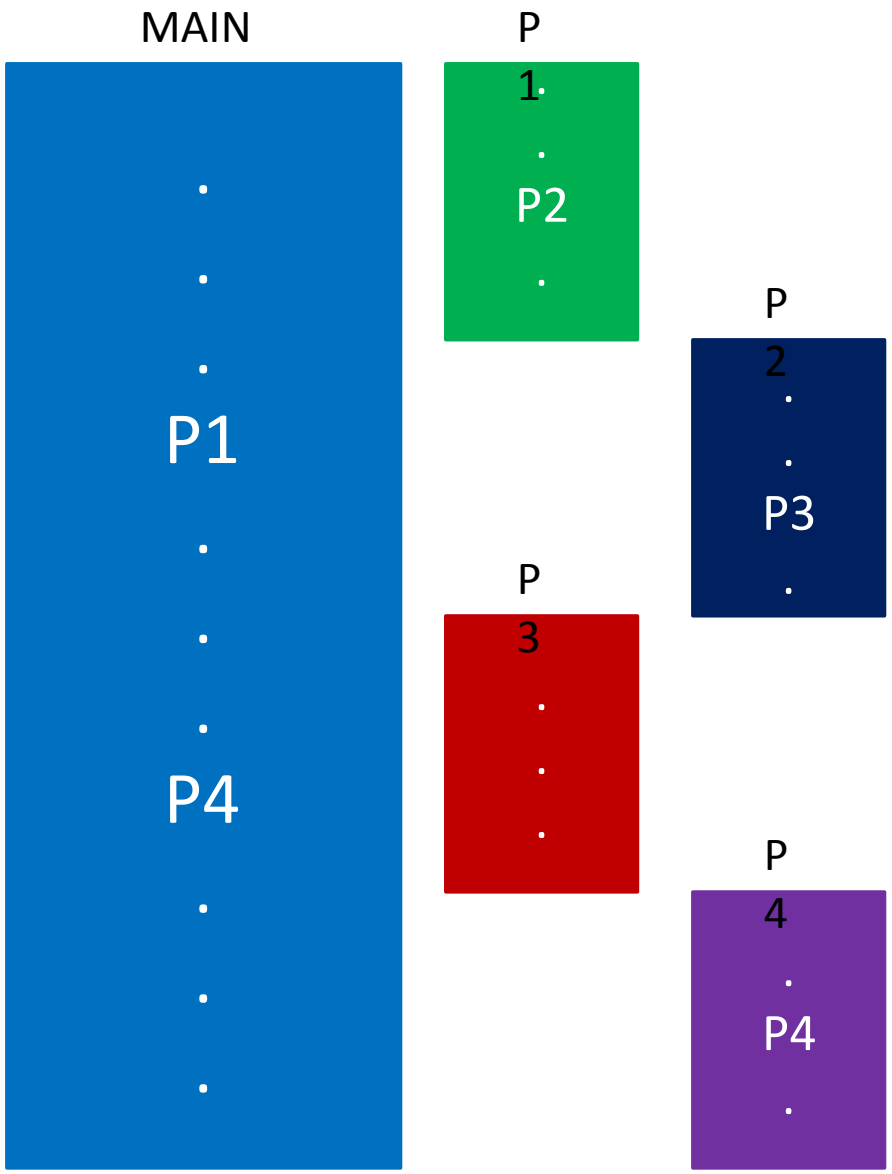
Assistant Professor
https://sites.google.com/view/mrstrusha

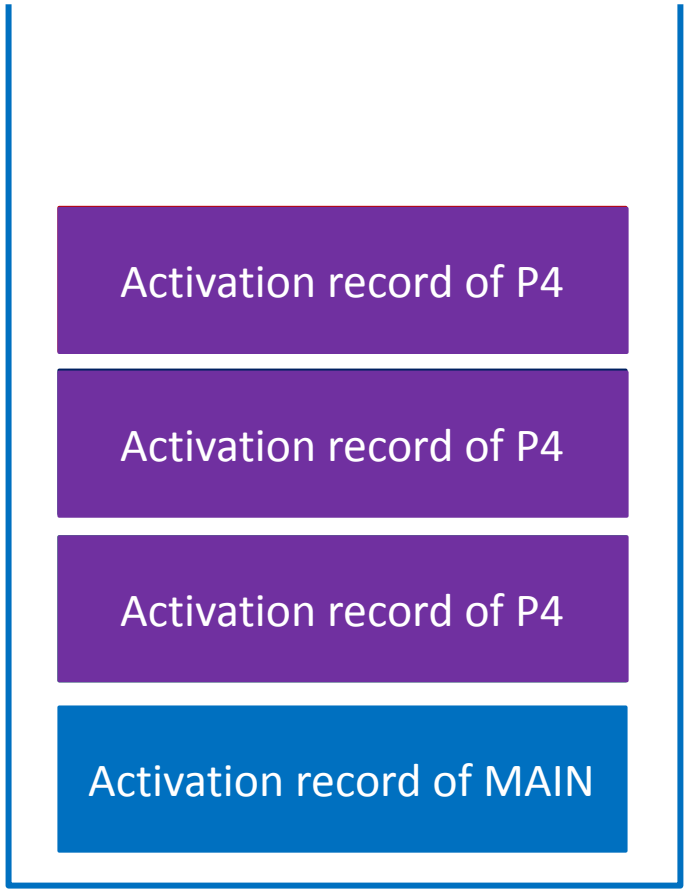

Subroutine
Procedure
Macro
Function


Main
IS
Procedure
CALL
Procedure
CALL
Procedure
CALL

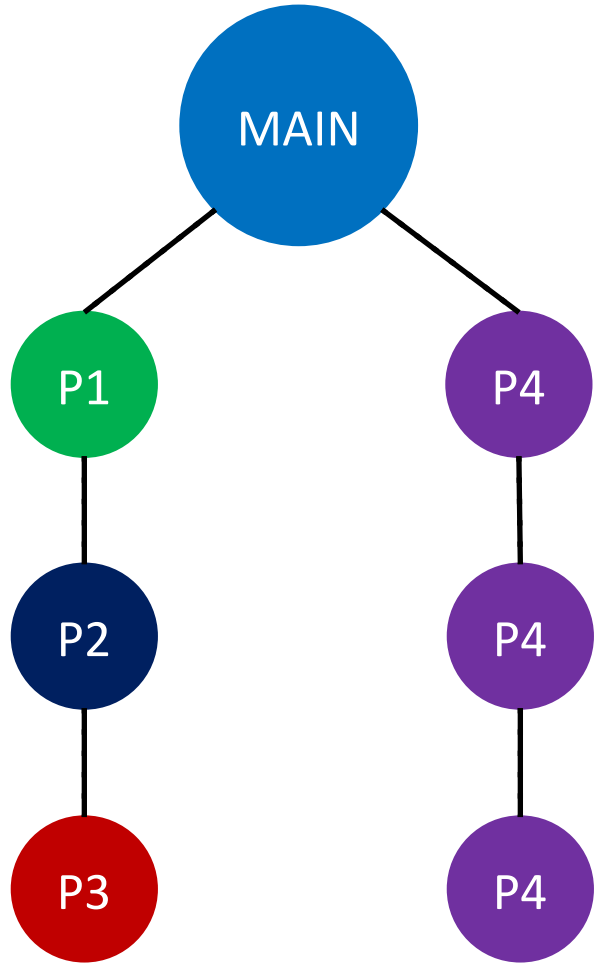

Execution of procedure
Refer
Activation of procedure
For each activation
Activation Record
Activation Record
Return value
Actual parameters
Control link
Access link
Saved machine status
Local data
Temporaries

Activation Tree	Depict the way control enters and leaves activation
Control Stack	Keep track of live procedure activation

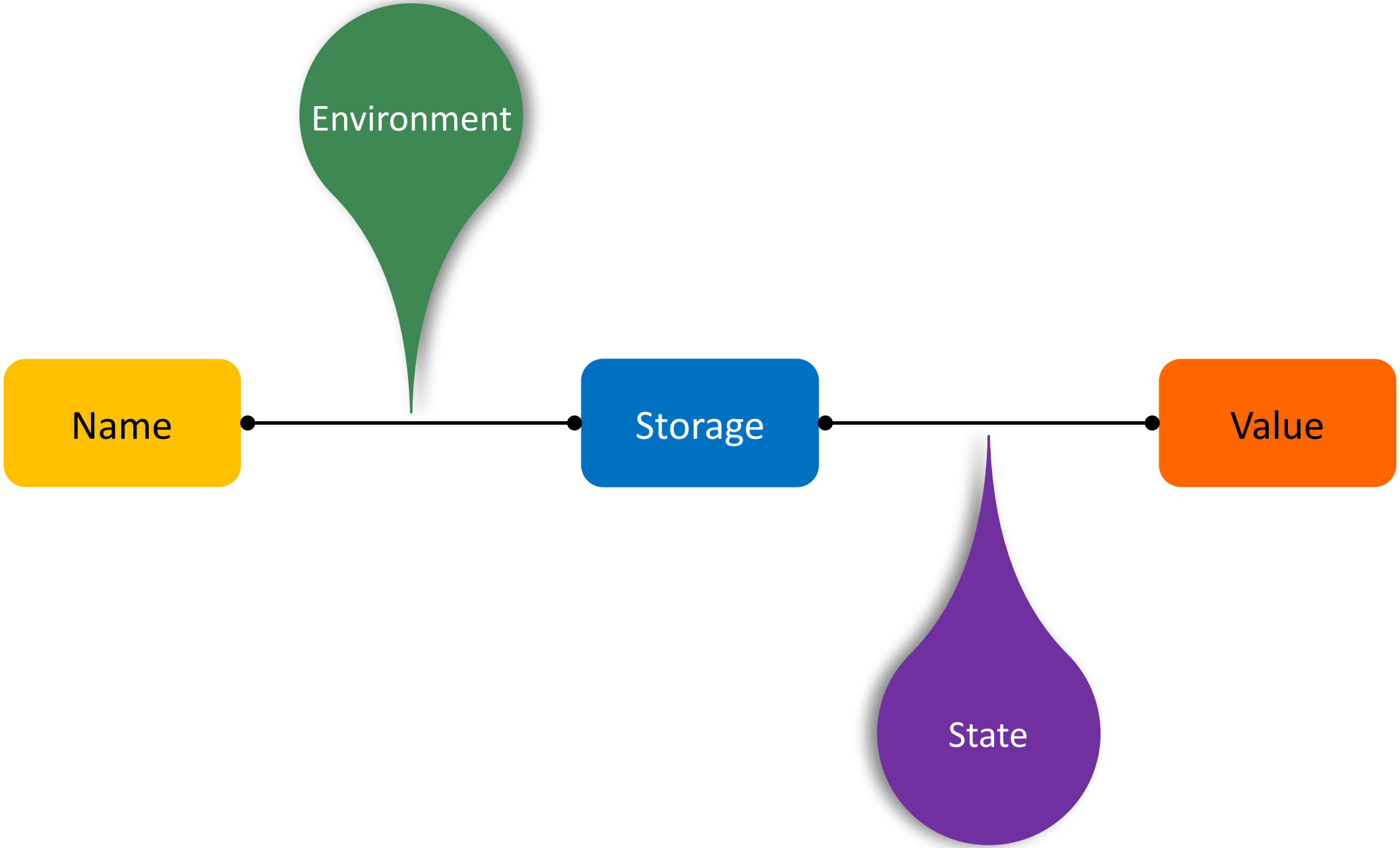


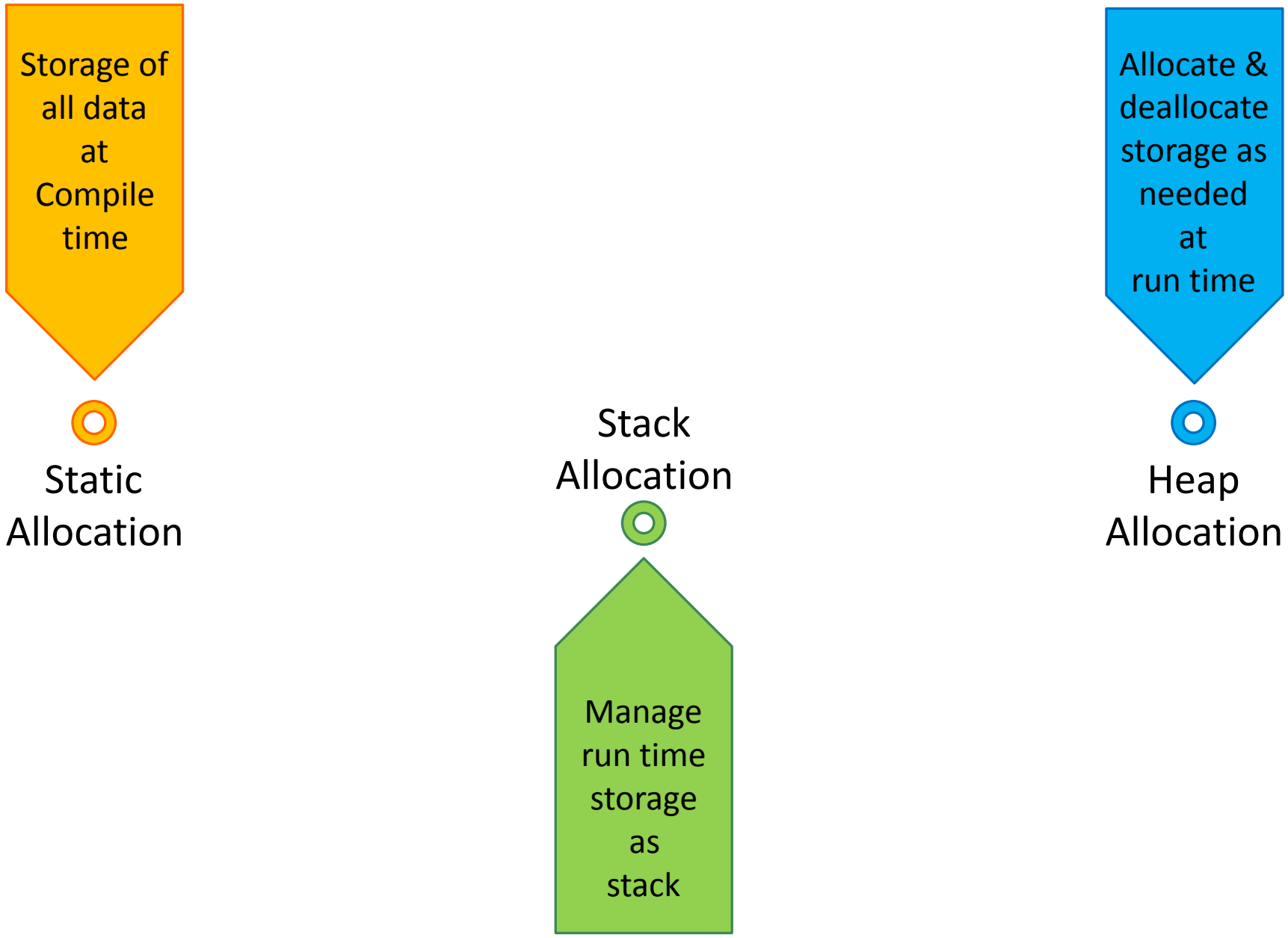


Control Stack



Activation Tree


Environment
Name
Storage
Value
State


Storage of all data at Compile time
Static Allocation
Stack Allocation
Manage run time storage as stack
Allocate & deallocate storage as needed at run time
Heap Allocation

Static Allocation

Names are bound to storage at the time of **compilation** of program

No need of run time support package

Binding do **not change** at **run time**

Every time a procedure is activated, names are bound to **same storage**

Value of local names **retained** across activation of procedure

Static Allocation

Size of data object must be **known** at compile time

Recursive procedures are **restricted**, as all activations use the same binding

Data structures cannot be created dynamically

Stack Allocation

Storage is organized as **stack**

Activation records are pushed and popped as activation begin and end

Locals are bound to **fresh storage** in **each activation**

Values of locals are **deleted** when the activation ends

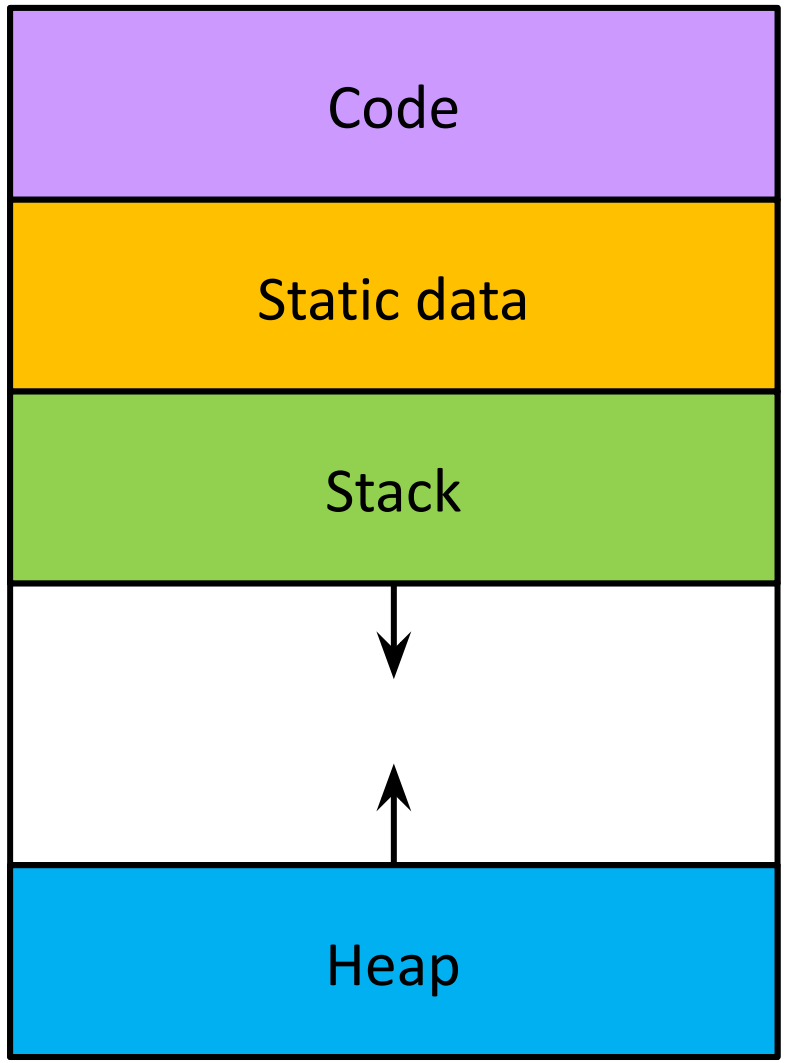
Size of data object must be **known** at compile time

Heap Allocation

Allocation and deallocation of storage at **run time**

Some time and **space overhead** associated with heap manager

Not required to **know** the **size** of data object at compile time


Code
Static data
Stack
Heap

Reference

- Alfred Aho, Ravi Sethi, Jeffrey D Ullman, *Compilers Principles, Techniques and Tools*, Pearson Education Asia.